CLAIMS

What is claimed is:

1. A method for constructing a software application from a model representation, the method comprising steps of:
   - reading the model representation;
   - identifying in the model representation a plurality of software modules;
   - identifying a programming language for each one of the plurality of software modules;
   - compiling each one of the plurality of software modules into machine language using a software compiler, the software compiler corresponding to the identified programming language; and
   - linking the compiled plurality of software modules into the software application.

2. The method of claim 1 wherein the step of reading the model representation further comprises the step of reading the model representation from one or more electronic files.

3. The method of claim 1 wherein the step of reading the model representation further comprises step of identifying a destination platform to the software application.

4. The method of claim 3 wherein the machine language is compatible with the identified destination platform.

5. The method of claim 3 wherein the destination platform comprises a processor architecture and an Operating System (OS).

6. The method of claim 4 wherein the OS can be one of Microsoft Windows™ based OS, UNIX ™ based OS, and Linux based OS, a real-time OS or a proprietary OS.

7. The method of claim 1 wherein the model representation is a graphical representation.

8. The method of claim 7 wherein the graphical representation conforms to the Unified Modeling Language (UML™) symbolic representation.

9. The method of claim 1 wherein the model representation is a textual representation.

10. The method of claim 1 wherein the identified programming language can be one of: Java™, C/C++, Ada, ALGOL, Assembly, COBOL, FORTRAN, Pascal, Perl, PL/I, Basic and family (Visual Basic™, Quick Basic™), PHP, ASP, Delphi™, SQL, CGI, XML, HTML, WAP or a proprietary programming language.

11. The method of claim 1 wherein the step of identifying a programming language for each one of the plurality of software modules further comprises step of generating the corresponding source code.

12. The method of claim 11 wherein the step of compiling each one of the plurality of software modules into machine language using a software compiler further comprises compiling the generated source code.

13. The method of claim 1 wherein a plurality of software compilers corresponding to the identified programming language are used.

14. The method of claim 1 wherein the software compiler is a complete independent software application.

15. The method of claim 1 wherein the software compiler is an incorporated software application.

16. A tool for constructing a software application, the tool comprising:

- an interpreting module for identifying a plurality of programming languages in a source code listing; and
- a calling module for compiling the source code listing into machine language.

17. The tool of claim 16 wherein the source code listing is contiguous or separated in multiple parts.

18. The tool of claim 16 wherein the calling module uses a plurality of appropriate software compilers for compiling each one of the plurality of programming languages.

19. The tool of claim 18 wherein at least one of the plurality of software compilers are complete independent software applications.

20. The tool of claim 18 wherein at least one of the plurality of software compilers is incorporated in the tool.